



Tipos de excepciones

Transcripción

[00:00] Vamos a hacer este este tópico un poco más detallado, un poco más elaborado porque este tema de las excepciones al inicio, cuando nunca lo has visto, es un poco complicado de entender. Un poco complicado, pero una vez que agarras la pegada, una vez que ya sabes de qué trata, todo vuela.

[00:22] Entonces nuevamente, recapitulando, para para ir poco a poco en orden, ya estamos viendo que debido a este error que salió aquí, que yo lo causé, es una división entre 0. Bueno, ustedes se preguntarán, pero a nadie en su sano juicio va a hacer una división entre 0 en un mundo real.

[00:44] Puede que no, puede que no, pero la mayoría de sistemas tienen, por lo general más de 100, 200, hasta 1000, 2000 archivos y hay veces que tú vas a reutilizar código, código, que no conoces, código que tú no has escrito. Ahí puedes darte con la sorpresa que hay referencias a null, como fue en el caso de testCuenta que yo hice una referencia a null y dio NullPointerException, y distintas excepciones de distintos tipos.

[01:13] Los errores no son malos, primera cosa que también tienen que entender es que los errores no son malos, son normales. No existe el código perfecto, siempre hay errores de todo tipo. Nuestro trabajo es aprender a lidiar con esos errores, aprender a manejar esos errores.

[01:30] Entonces, analizando el flujo nuevamente aquí vamos a usar nuestro debug. No sé si ya tenemos el punto aquí y iniciamos nuestro debug. La perspectiva cambia un poquito, perfecto, entro aquí, y le voy a decir que hag

step over, le voy a decir que entre a método 1. Perfecto, step over, y ahora step in, para entrar al método 2. Vemos aquí como él va pintando y aquí ya se va llenando mi stack.

[02:00] Hasta aquí ya tengo main, método 1, método 2. Y él ya va a imprimir Inicio método 2, luego de step over, él ya va a comenzar a iterar y él va a imprimir la primera línea. Ahora él va a declarar una variable num igual a cero y el resultado va a ser la división, vemos claramente cómo aquí sigue el stack lleno. Y en eso. ¿Qué sucede?

[02:25] Alguna cosa pasó porque él no salió de ahí. Y vemos que él dice Disconnected Terminated. ¿Qué pasó? Imprimió el lenguaje aquí y una cosa interesante que tiene que ver aquí, el mensaje que está dando la excepción. ¿Por qué? Porque porque él dice que ocurrió una excepción en el flujo main del tipo Java.lang.ArithmeticException: / by zero.

[02:54] ¿Dónde? En Flujo.método2(Flujo.java:21). ¿Donde fue llamado eso? En Flujo.método1(Flujo.java:12) y esto fue llamado en la línea 6, la línea 6 es la que está aquí que llama método 1. La línea 12 es la que de aquí que llama método 2 y la línea 21 es la que, en efecto, causa la excepción.

[03:17] Entonces, una cosa más que tenemos que aprender con el tema de excepciones es a leer el mensaje. Casi siempre la detección del error va a depender de qué tan bien sepamos entender el mensaje de la excepción, es muy importante saber leer el mensaje, saber las partes que tiene este mensaje.

[03:33] La primera parte es el nombre de la excepción. La primera parte es qué excepción, lanzó. La segunda parte es el mensaje. Hay excepciones que te da un mensaje sobre exactamente qué fue lo que ocurrió, qué es lo que pasa aquí. División por cero. La tercera parte es lo que llamamos el stack trace o en español, lo que sería pues la traza de la pila, lo que la pila fue recorriendo, y dejó aquí como unos logs, prácticamente, nuestro stack trace.

[04:08] Y aquí vemos claramente la pila que estaba aquí funcionando, método 2, método 1 y main. Básicamente entonces, volviendo a nuestra analogía de la bomba, lo que pasó fue que aquí explotó la bomba y simplemente nadie pudo desactivarla. Aquí salió la bomba y el programa dijo: "Yo no sé cómo trabajar con esa bomba, explotó y el programa pues fue exterminado, no pudo terminar su ciclo". Vamos a analizar un poco más gráficamente esto en el siguiente video.